```
procedure INIT-PDA
{Invoked when the router comes up. }
begin
   Initialize all tables;
   call PDA;
end INIT-PDA



procedure PDA
{Executed at each router i. Invoked when an event occurs}
begin
   (1) call NTU;
   (2) call MTU; /* Updates $T^i$ */
   (3) if (there are changes to $T^i$) then
             Compose an LSU message consisting of topology
             differences using add, delete
             and change link entries;
        end if
   (4) Within a finite amount time, send the
        LSU message to all neighbors;
end PDA
```

FIG. 1

```
procedure NTU
begin
   (1) if (LSU message is received from a neighbor $k$) then
            (1a) Update neighbor table $T^i_k$. That is, add links,
                   delete links or change links according to the
                   specification of each entry in the LSU;
            (1b) Run Dijkstra's shortest path algorithm
                   on the resulting topology $T^i_k$; /*This results in
                   finding minimum distances from $k$ to all other
                   nodes in $T^i_k$. Note $T^i_k$ is a tree*/
            (1c) Update $D^i_{jk}$ with new distances in $T^i_k$;
       end if
   (2) if (adjacent link $(i, k)$ is up) then
            Update $l^i_k$ and send an LSU message to the
            neighbor $k$ with link information of all links in
            its main topology table $T^i$;
       endif
   (3) if (cost of an adjacent link $(i, k)$ changed)then
             Update $l^i_k$;
       endif
   (4) if (adjacent link $(i, k)$ failed)then
             Update $l^i_k$ and clear the table $T^i_k$;
       endif
end NTU
```

FIG. 2

```
procedure MTU at router $i$
begin
        (1) $oldT^i \leftarrow T^i$:    /* Save copy */
        (2) if (node $j$ occurs in at least one of $T_k^i$) then
                add $j$ to the main topology table $T^i$
            end if
        (3) for each node $j$ in $T^i$ do
              $MIN \leftarrow min\ \{D_{jk}^i + l_k^i \mid k \in N^i\}$;
              let $p$ be such that $MIN = (D_{jp}^i + l_p^i)$;
              /* Neighbor $p$ is the preferred neighbor for
              destination $j$. Ties are broken in favor of
              lower address neighbor */
            done
        (4) for each $j$ in $T^i$ and its preferred neighbor $p$ do
                Copy all links $(j,n)$ from $T_p^i$ to $T^i$;
                /* i.e., copy all links in $T_p^i$ for which
                $j$ is the head node */
             done
        (5) Update $T^i$ with information of each $l_k^i$;
        (6) Run Dijkstra's shortest path algorithm on $T^i$
             and remove those links in $T^i$ that are *not*
             part of the shortest path tree;
        (7) Update $D_j^i$ with new distances in $T^i$;
        (8) Compare $oldT^i$ with $T^i$ and note all differences;
end MTU
```

FIG. 3

```
procedure MPDA at router $i$
{$invoked\ when\ an\ event\ occurs$}
begin
    (1) call NTU;
    (2) if (node is in PASSIVE state) then
        (2a) call MTU; /* update $T^i$and $D_j^i$ */
        (2b) $F\,D_j^i \leftarrow min\{F\,D_j^i, D_j^i\}$;
      endif
    (3) if (node is in ACTIVE state and the
            last ACK is received) then
        (3a) $temp_j^i \leftarrow D_j^i$; set node to PASSIVE state;
        (3b) call MTU to update $T^i$;
        (3c) $F\,D_j^i \leftarrow min\{\ temp_j^i, D_j^i\}$
      endif
    (4) $S_j^i \leftarrow \{k | D_j^i < F\,D_j^i\}$;
    (5) if (changes occur in $T^i$)then
        Set node to ACTIVE state;
      endif
      if (no changes occur in $T^i$ and the event is
            the last ACK) then
        Set node to PASSIVE state;
      endif
    (6) if (there are changes to $T^i$) then
         Compose anew LSU with the topology
         changes expressed as *add* link,
         *delete* link and *change* link;
      end if
    (7) if (input event received is an LSU message)then
        Add the ACK entry to newly composed LSU;
      endif
    (8) Send the new LSU message.
end MPDA
```

FIG. 4

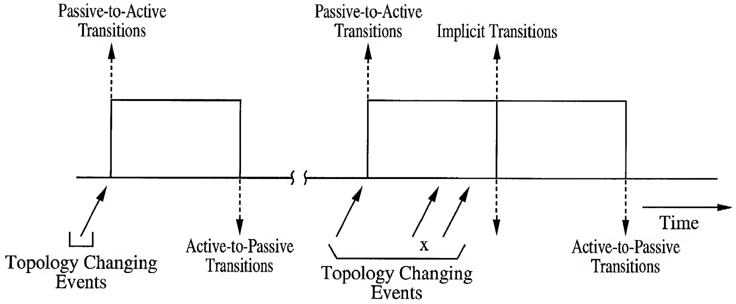
Passive-to-Active Transitions
Passive-to-Active Transitions
Implicit Transitions
Topology Changing Events
Active-to-Passive Transitions
x
Topology Changing Events
Time
Active-to-Passive Transitions

FIG. 5

```
Procedure IH
begin
  (1) $\forall k \notin S^i_{j,} \phi^i_{jk} \leftarrow 0;$
  (2) if $(|S^i_j| = 1)$ then
        $\forall k \in S^i_{j,} \phi^i_{jk} \leftarrow 1;$
        endif
  (3) if $(|S^i_j| > 1)$ then
        $\phi^i_{jk} \leftarrow \dfrac{1 - \dfrac{D^i_{jk} + l^i_k}{\sum_{m \in S^i_j} (D^i_j + l^i_m)}}{(|S^i_j| - 1)}, \forall k \in S^i_j;$
        endif
end IH
```

FIG. 6

```
Procedure AH
begin
    (1) $D^{ij}_{min} \leftarrow min\ \{D^{i}_{jk} + l^{i}_{k} | k \in S_j\}$ ;
    (2) let $D^{ij}_{min} = (D^{i}_{jk_0} + l^{i}_{k_0})$
        // that is, $k_0$ be the neighbor
        that offers the minimum)
    (3) foreach $k \in S^{i}_{j}$ do
            $a^{i}_{jk} \leftarrow D^{i}_{jk} + l^{i}_{k} - D^{ij}_{min}$;
        done
    (4) $\Delta \leftarrow \frac{1}{2} min\ \{\frac{\phi^{i}_{jk}}{a^{i}_{jk}} | k \in S^{i}_{j} \wedge a^{i}_{jk} \neq 0\}$;
    (5) foreach $k \neq k_0 \wedge k \in S^{i}_{j}$ do
            $\phi^{i}_{jk} \leftarrow \phi^{i}_{jk} - \Delta \times a^{i}_{jk}$;
        done
    (6) foreach $k = k_0$ do
            $\phi^{i}_{jk} \leftarrow \phi^{i}_{jk} + \sum_{q \in S_{jk}} \Delta \times a^{i}_{jq}$;
        done
    end AH
```

FIG. 7

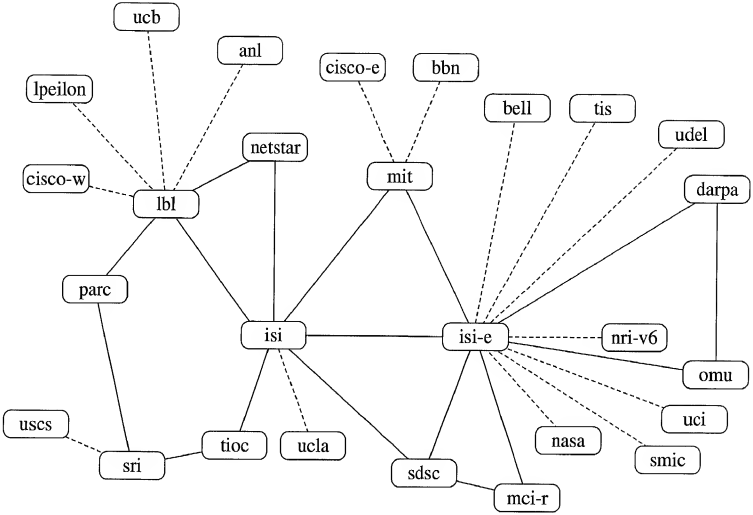
ucb
anl
lpeilon
cisco-e
bbn
bell
tis
udel
netstar
cisco-w
lbl
mit
darpa
parc
isi
isi-e
nri-v6
omu
uci
uscs
sri
tioc
ucla
sdsc
nasa
smic
mci-r

FIG. 8

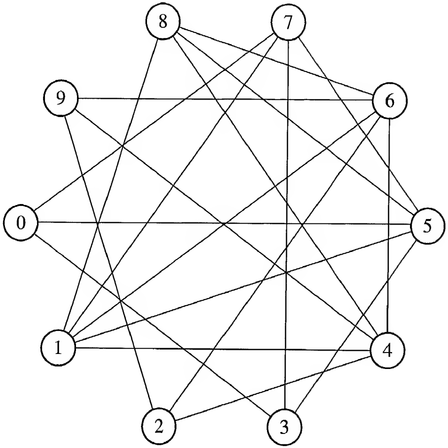

FIG. 9

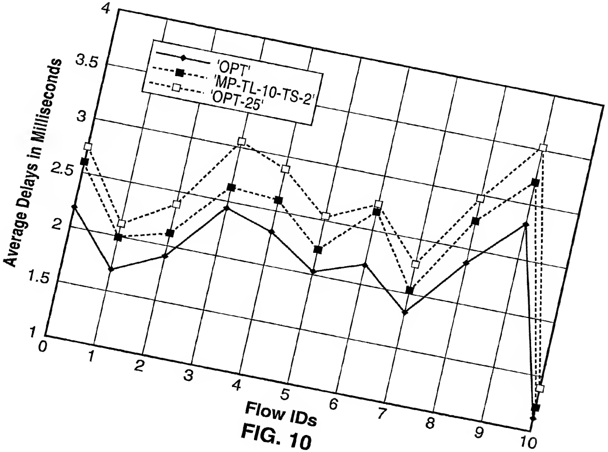
'OPT'
'MP-TL-10-TS-2'
'OPT-25'
Average Delays in Milliseconds
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9
10
Flow IDs

FIG. 10

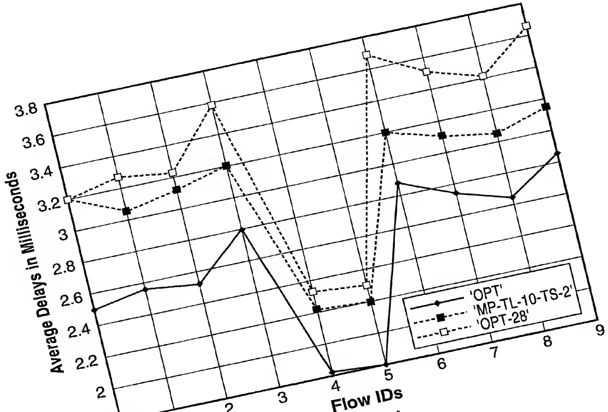
Average Delays in Milliseconds
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
0
1
2
3
4
5
6
7
8
9
Flow IDs
'OPT'
'MP-TL-10-TS-2'
'OPT-28'

FIG. 11

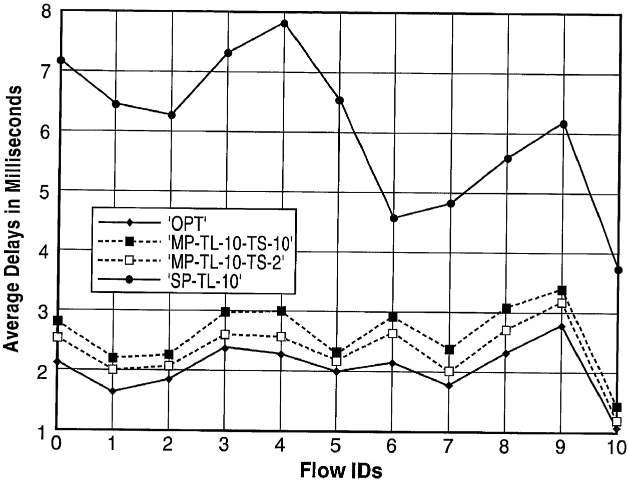
Average Delays in Milliseconds
8
7
6
5
4
3
2
1
0
1
2
3
4
5
6
7
8
9
10
Flow IDs
'OPT'
'MP-TL-10-TS-10'
'MP-TL-10-TS-2'
'SP-TL-10'

FIG. 12

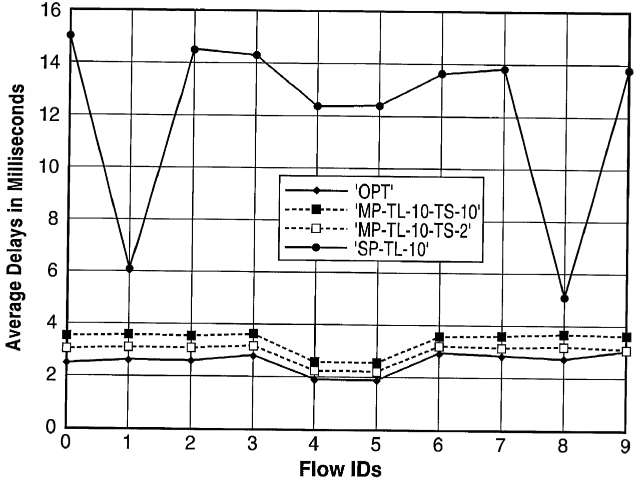
Average Delays in Milliseconds
16
14
12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
9
Flow IDs
'OPT'
'MP-TL-10-TS-10'
'MP-TL-10-TS-2'
'SP-TL-10'

FIG. 13

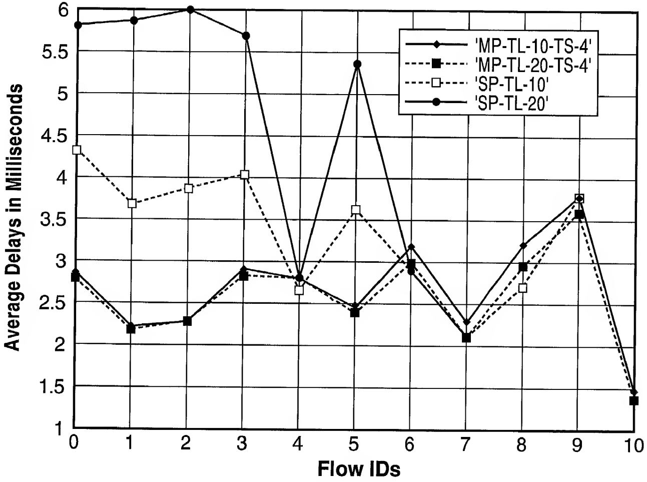
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
Average Delays in Milliseconds
0
1
2
3
4
5
6
7
8
9
10
Flow IDs
'MP-TL-10-TS-4'
'MP-TL-20-TS-4'
'SP-TL-10'
'SP-TL-20'

FIG. 14

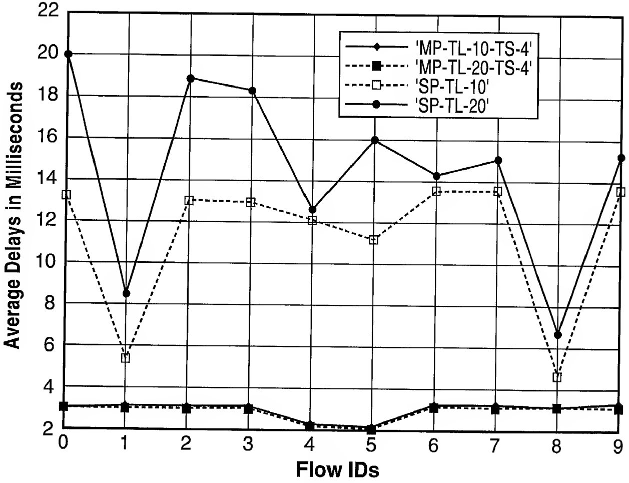
22
20
18
16
14
12
10
8
6
4
2
Average Delays in Milliseconds
0
1
2
3
4
5
6
7
8
9
Flow IDs
'MP-TL-10-TS-4'
'MP-TL-20-TS-4'
'SP-TL-10'
'SP-TL-20'

FIG. 15

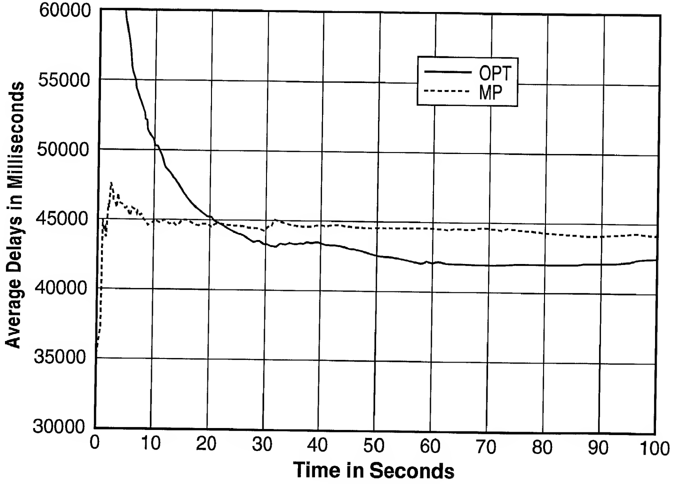
60000
55000
50000
45000
40000
35000
30000
Average Delays in Milliseconds
0
10
20
30
40
50
60
70
80
90
100
Time in Seconds
OPT
MP

FIG. 16

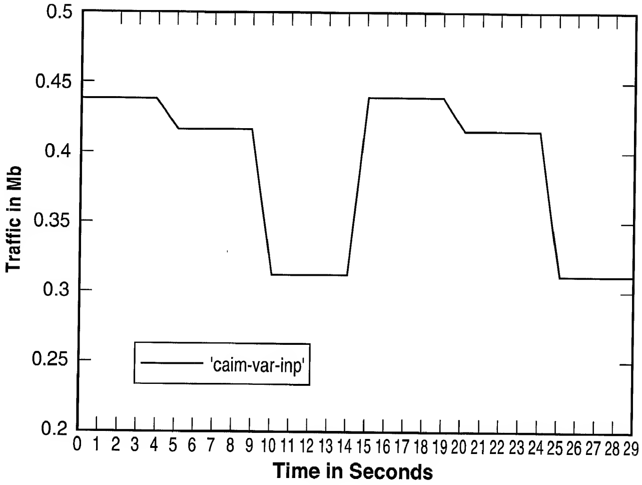
0.5
0.45
0.4
0.35
0.3
0.25
0.2
Traffic in Mb
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Time in Seconds
'caim-var-inp'

FIG. 17

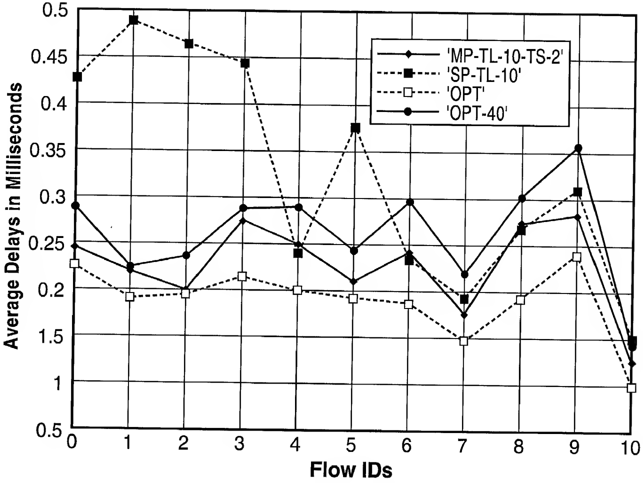
Average Delays in Milliseconds
0.5
0.45
0.4
0.35
0.3
0.25
0.2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
Flow IDs
'MP-TL-10-TS-2'
'SP-TL-10'
'OPT'
'OPT-40'

FIG. 18

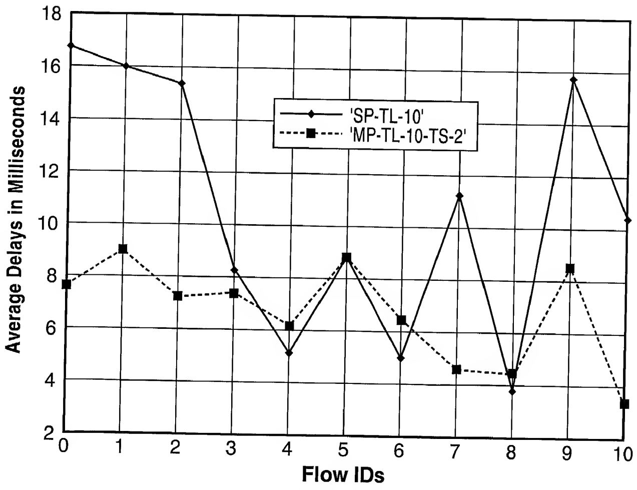
'SP-TL-10'
'MP-TL-10-TS-2'
Average Delays in Milliseconds
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
Flow IDs

FIG. 19